



# Fundamentos de Programação 2

## Curso Técnico de Informática – Departamento de Telemática

**Prof. César Olavo de Moura Filho, PhD**

# Fundamentos de Programação 2

**Porque se dedicar a este curso:**

## **1) Veremos Javascript**

- a propósito, alguém aqui já programa Javascript?

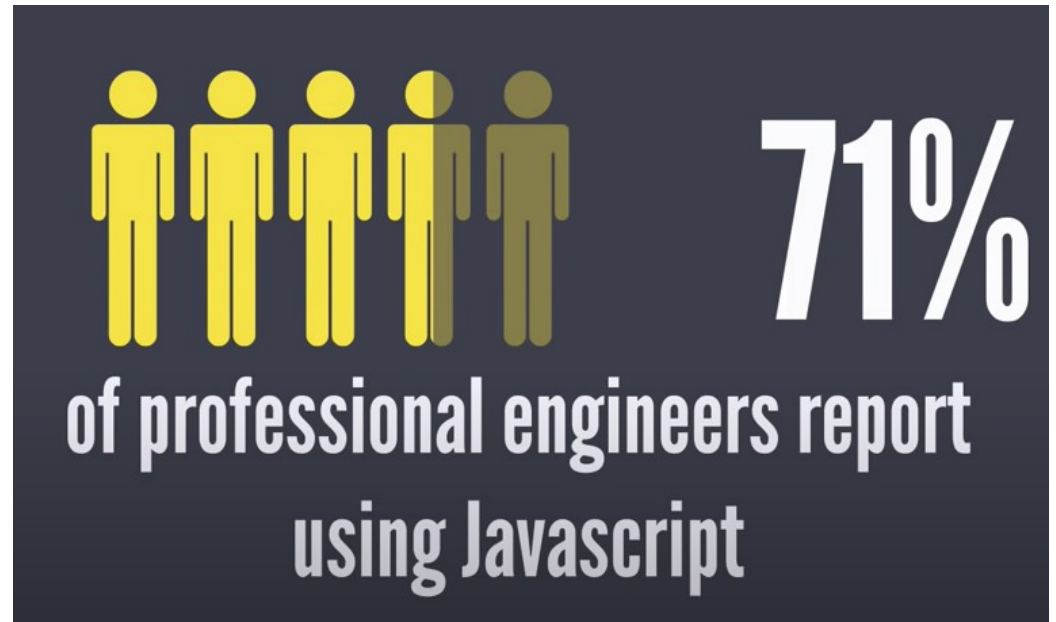
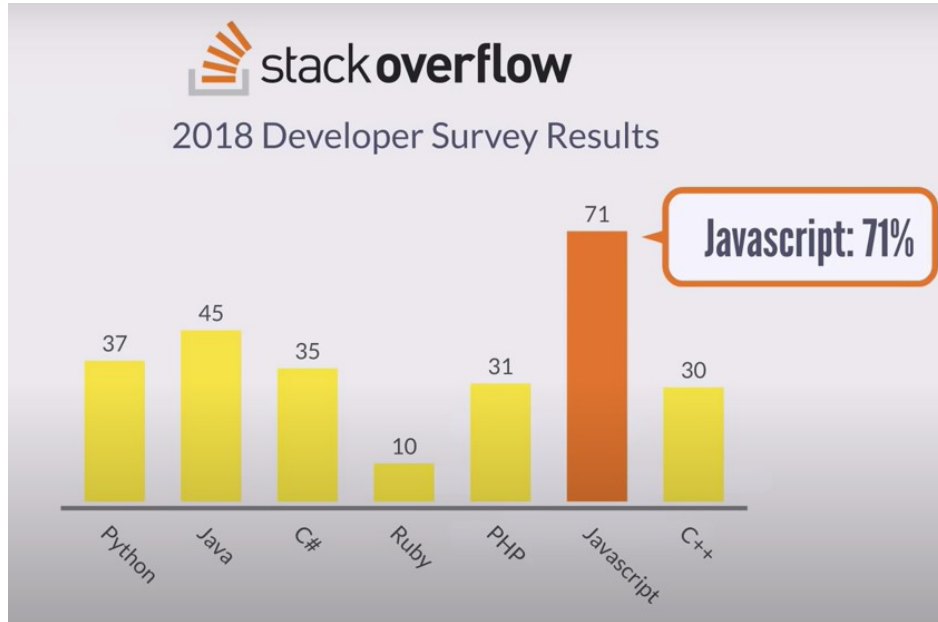
## **2) Ouvi alguém perguntar porque Javascript?**

- Linguagem da WEB: padrão de desenvolvimento Web (presente em 95% dos websites!)
- Programação Web 1 (P5) e Programação Web 2 (P6)

## **3) Sugiro fortemente continuar estudando Javascript, mesmo no intervalo (P4)**

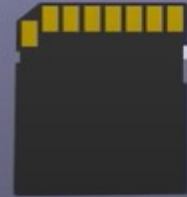
# Javascript é popular?

Evolução das linguagens de programação: <https://www.youtube.com/watch?v=qogEJSNZTPA>



**Javascript: usado para que?**

# Javascript is everywhere



# Javascript

- JavaScript (às vezes abreviado para JS) é uma linguagem *interpretada e baseada em objetos com funções de primeira classe*.
- Mais conhecida como uma linguagem de script para páginas Web, mas usada também em outros ambientes sem browser, tais como node.js.
- O JavaScript é uma linguagem baseada em protótipos, multi-paradigma e dinâmica, suportando estilos de orientação a objetos, imperativos e declarativos (como por exemplo a programação funcional).
- Javascript não é Java!
- Como dissemos, é o padrão de desenvolvimento Web

# Desenvolvimento Web

Sem o Javascript, a Web seria uma aplicação “burra”, pois não seriam possíveis várias facilidades a que ficamos acostumados:

- Compras online
- Redes sociais
- Trocas de mensagens
- Compartilhar arquivos, fotos, etc.
- Reservar bilhetes, hotéis, etc.

Ou seja, graças ao Javascript, essas aplicações que já existiam, mas residiam no PC, computadores, etc. vieram para a Web.

# Web pré-Javascript



The screenshot shows the classic Yahoo! homepage layout. At the top is the "YAHOO!" logo in large red letters. To the left of the logo are icons for "What's New" and "Check Email". To the right are icons for "Personalize" and "Help". Below the logo, there's a section for "Yahoo! Mail" with the text "free email for life". To the right of that is a "Know when friends are online!" section with a "Click to download Yahoo! Messenger" button. Below these is a search bar with a "Search" button and a link to "advanced search". To the left of the search bar is a "Shopping" link. Below the search bar is a horizontal menu with links: Shopping - Auctions - Yellow Pages - People Search - Maps - Travel - Classifieds - Personals - Gay Mail - Calendar - Messenger - Companion - My Yahoo! - News - Sports - Weather - TV - Stock. Below this menu is a "Yahoo! Shopping" section with the text "Thousands of stores. Millions of products." and a table of departments, stores, and products. To the right of the shopping section is a "In the" section with a list of links. Below the shopping section is a "Arts & Humanities" section with links to Literature, Photography, etc. To the right of that is a "News & Media" section with links to Full Coverage, Newspapers, TV, etc. Below the arts section is a "Business & Economy" section with links to Companies, Finance, Jobs, etc. To the right of that is a "Recreation & Sports" section with links to Sports, Travel, Autos, Outdoors, etc. Below the business section is a "Computers & Internet" section with links to Internet, WWW, Software, Games, etc. To the right of that is a "Reference" section with links to Libraries, Dictionaries, Quotations, etc. Below the computers section is an "Education" section with links to College and University, K-12, etc. To the right of that is a "Regional" section with links to Countries, Regions, US States, etc. Below the education section is an "Entertainment" section with links to Cool Links, Movies, Humor, Music, etc. To the right of that is a "Science" section with links to Animals, Astronomy, Engineering, etc.

**YAHOO!**

What's New Check Email Personalize Help

**Yahoo! Mail**  
free email for life

Know when friends are online!  
Click to download Yahoo! Messenger

Search advanced search

Shopping - Auctions - Yellow Pages - People Search - Maps - Travel - Classifieds - Personals - Gay Mail - Calendar - Messenger - Companion - My Yahoo! - News - Sports - Weather - TV - Stock

**Yahoo! Shopping** - Thousands of stores. Millions of products.

Departments	Stores	Products
<a href="#">Apparel</a>	<a href="#">Sports Authority</a>	<a href="#">Digital cameras</a>
<a href="#">Bath/Beauty</a>	<a href="#">Gap</a>	<a href="#">Pokemon</a>
<a href="#">Computers</a>	<a href="#">Eddie Bauer</a>	<a href="#">MP3 players</a>
<a href="#">Electronics</a>	<a href="#">Macy's</a>	<a href="#">DVD players</a>

**Arts & Humanities**  
[Literature](#), [Photography](#),...

**Business & Economy**  
[Companies](#), [Finance](#), [Jobs](#),...

**Computers & Internet**  
[Internet](#), [WWW](#), [Software](#), [Games](#),...

**Education**  
[College and University](#), [K-12](#),...

**Entertainment**  
[Cool Links](#), [Movies](#), [Humor](#), [Music](#),...

**News & Media**  
Full Coverage, [Newspapers](#), [TV](#),...

**Recreation & Sports**  
[Sports](#), [Travel](#), [Autos](#), [Outdoors](#),...

**Reference**  
[Libraries](#), [Dictionaries](#), [Quotations](#),...

**Regional**  
[Countries](#), [Regions](#), [US States](#),...

**Science**  
[Animals](#), [Astronomy](#), [Engineering](#),...

**In the**

- [Scores killer riots](#)
- [Austria's Ha party leader](#)
- [Floods trap Mozambique](#)

**Market**

- [Y! Auctions](#)
- [Pokemon, etc](#)
- [Free 56K In](#)
- [Yahoo! Bill month trial](#)

**Inside**

- [Yahoo! Geo your free ho](#)
- [Play free Fa](#)
- [Yahoo! Club](#)



The screenshot shows the website for the Escola Técnica Federal do Ceará (ETFCE). At the top is a banner image of the school's building with the text "ESCOLA TÉCNICA FEDERAL DO CEARÁ". To the right of the banner is a small image of a gear and the text "topmiddleright2.jpg (12800 bytes)". Below the banner is a navigation menu with links: Escola, Cursos, Pesquisa, Serviços, Eventos, Publicações, and Relações comunitárias. To the right of the menu is a contact section with a phone number "888888" and the text "Dúvidas ou sugestões: adrianom@etfce.br".

topmiddleright2.jpg  
(12800 bytes)

register help

**ESCOLA TÉCNICA FEDERAL DO CEARÁ**

888888

Dúvidas ou sugestões:  
[adrianom@etfce.br](mailto:adrianom@etfce.br)

**Escola**

**Cursos**

**Pesquisa**

**Serviços**

**Eventos**

**Publicações**

**Relações comunitárias**

# Bibliografia

MDN:

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

W3schools:

<http://w3schools.com>

Google: 

<https://google.com>



# Comparando JS com outras linguagens: Java

Quem vem do Java e se propõe a estudar JS, vai se sentir muito desconfortável:

Interpretada



Como executar essa \$%@#% de programa?



Single-threaded (**assíncrona**? callback? promise?)



Tipagem dinâmica



Função: cidadã de primeira classe




Herança por Protótipo



# Comparando JS com outras linguagens: python

Quem vem do Python, é (um pouco) mais tranquilo:

Interpretada 

Como executar essa \$%#@#% de programa? 

No REPL for you!

Single-threaded (**assíncrona**? callback? Promise?)   






Tipagem Dinâmica 

Função cidadã de primeira classe 

Herança por Protótipo 

# Comparando JS com outras linguagens: python

**Outros detalhes menos importante que diferenciam Javascript de python (e que veremos nas práticas):**

- Hoisting 
- Bloco de código com chaves 
- Ponto-e-vírgula 
- Arrow function 
- Chamada de função com parâmetros exatos 

# Interpretada?

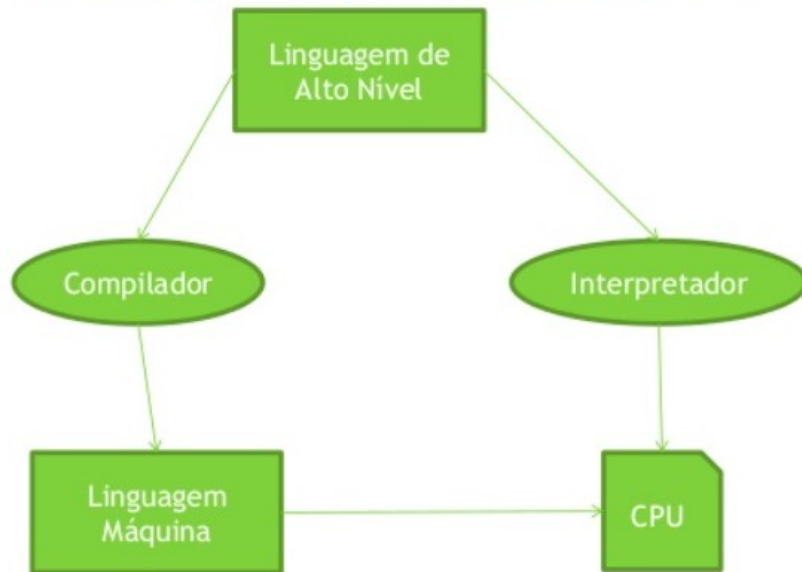
## Interpretada vs. compilada

Não é inerente a uma linguagem

**Interpretada:** o código é transformado em código de máquina e executado em tempo de execução. Linguagens interpretadas são transformadas em código de máquina toda vez que um script precisa ser executado.

Linguagens **compiladas** precisam ser transformadas em código de máquina apenas uma vez (durante a compilação)

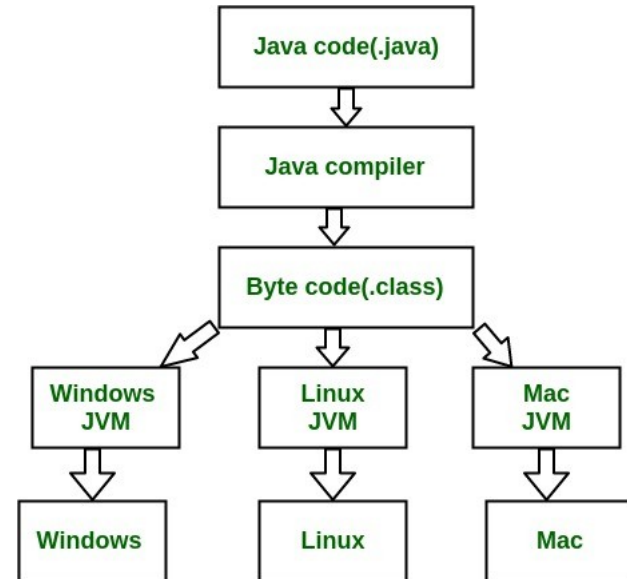
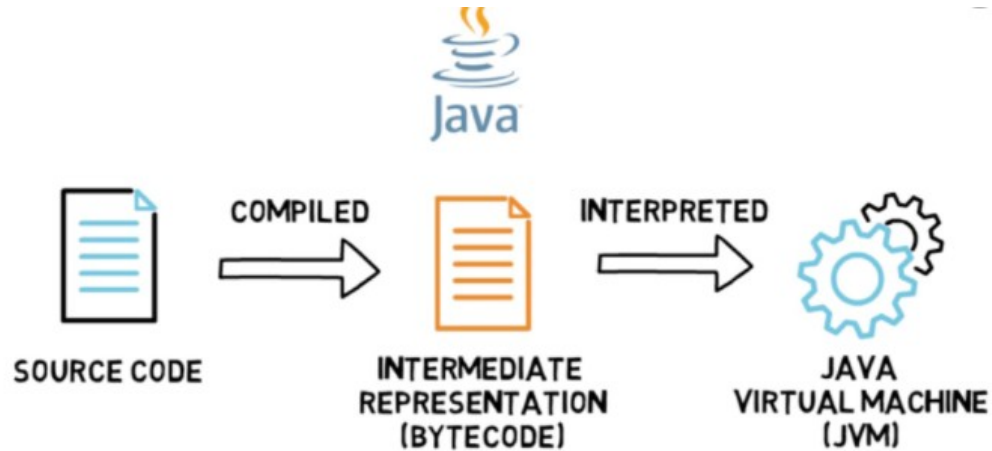
## COMPILADOR E INTERPRETADOR



# Interpretada?

Mas nada em informática é tão simples assim:

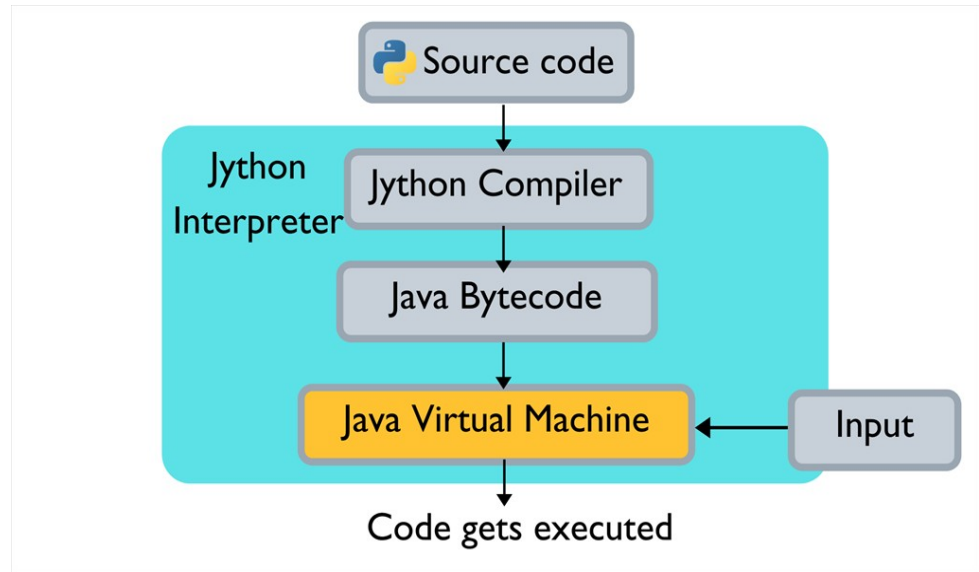
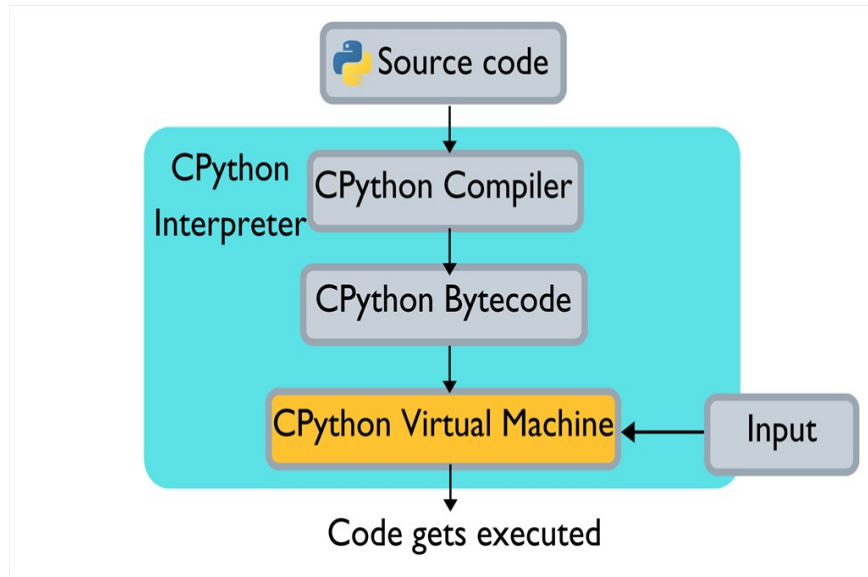
Just in Time (JiT) vs Ahead of Time (AoT)



# Interpretada?

## Python

Várias implementações: CPython, Jython, PyPy



# Interpretada? Javascript

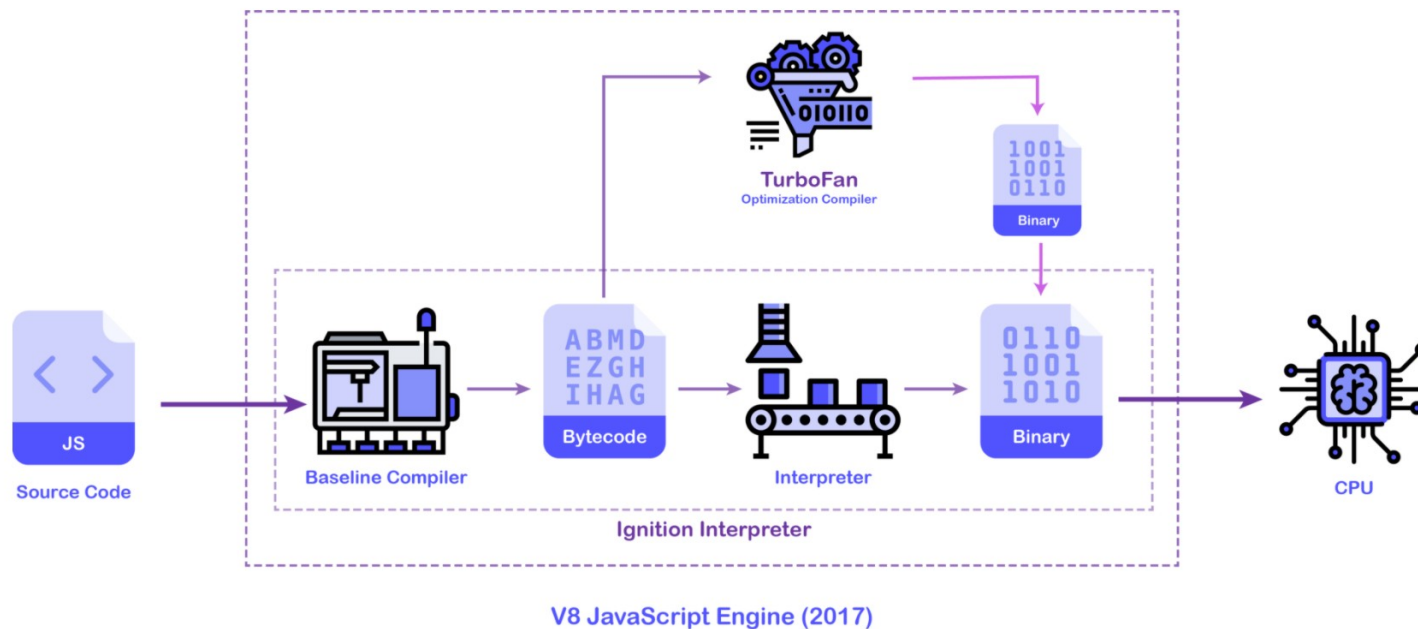
Como todo interpretador moderno, envolve compilação também (JIT)

Várias engines:

V8

SpiderMonkey

Rhino



## Onde executar?

No Javascript há algumas maneiras de se executar um programa, algumas para fins de teste apenas e outras para a programação:

- a) console dos browsers (Chrome, Edge e Firefox)
- b) node.js (via REPL ou executando um módulo)
- c) plugins/extensões de alguns editores (e.g. vscode)
- d) interpretadores online (codepen, glitch, replit, etc)
- e) browsers (numa página HTML ou por ela referenciada)



## Onde executar? (ambientes de experimentação)

**Inicialmente, para fins didáticos, usaremos o console de algum browser para executar códigos:**

Chrome/Edge/Firefox: ctrl+shift+i (ou F12)

Não salva.

**Além do console dos browsers, há a alternativa dos editores online:**

repl.it,

codepen.io,

glitch.com, etc.

## Onde executar? (ambientes de programação)

- Os ambientes do slide anterior são adequados apenas para experimentos e debug
- Em ambiente de produção, os programas JavaScript esperam por ações do usuário (chamadas **eventos**) e a elas respondem. Nesse caso, temos dois principais ambientes para executar nossos programas:
  - O browser (*não* o console do browser)
  - O Node.JS
- A seguir faremos o famoso “Alô, mundo” em ambos os ambientes (prática).

# Javascript no Browser

- Adicione no seu arquivo HTML a tag <script>
- Insira na tag acima todo seu código Javascript

```
1 <html>
2   <script>
3     console.log("alô, mundo");
4   </script>
5 <body>
6   
7 </body>
8 </html>
```

```
1 <html>
2   <script src="javascript/app.js">
3
4   </script>
5 <body>
6
7 </body>
8 </html>
```

# Exercícios

Pesquise sobre as seguintes características da linguagem Javascript:

- *Single-threaded*
- *Event loop*
- *Tipagem dinâmica vs. tipagem estática*
- *Herança por protótipo (vs. Herança por classe)*
- *Hoisting*

*Ps: nos slides a seguir encontram-se algumas informações sobre os temas acima e que podem ajudar na sua pesquisa*

# Single Threaded?

## Mas o que é um *thread* mesmo?

Digamos que você saiba o que é um processo:

Threads são processos leves que podem ser executados em paralelo e compartilhar um espaço de memória e outros recursos com seus processos pais.

Tem a ver com:

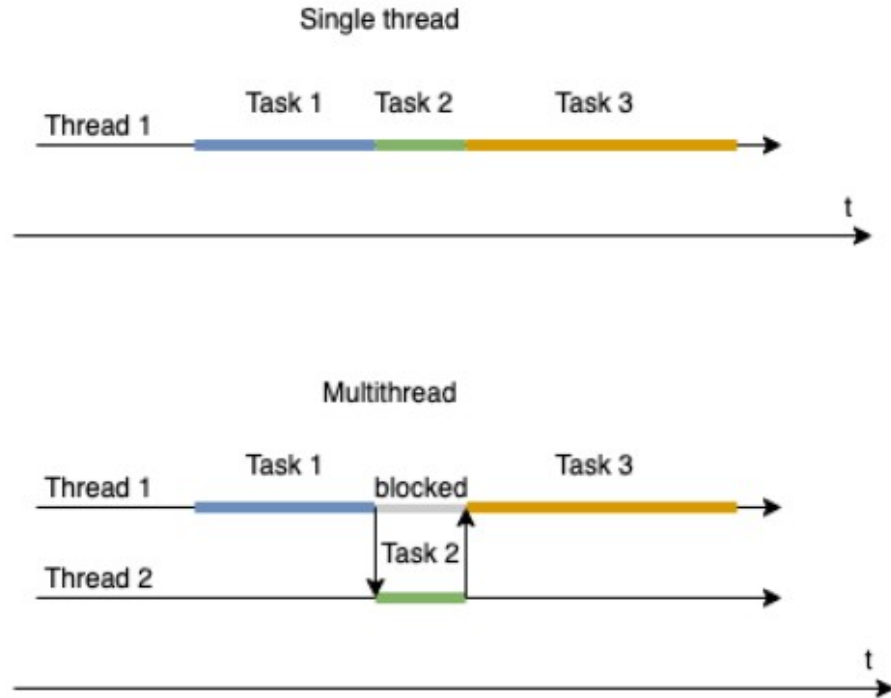
Síncrono vs assíncrono

Paralelo vs. sequencial

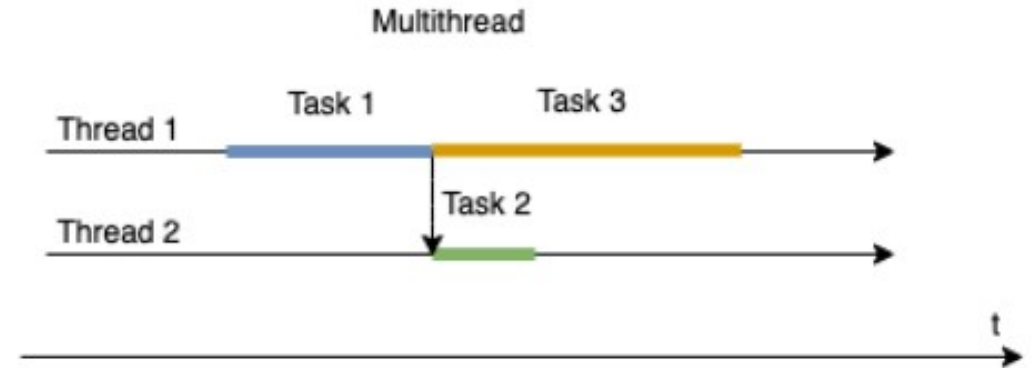


# Single Threaded?

## Síncrono



## Assíncrono

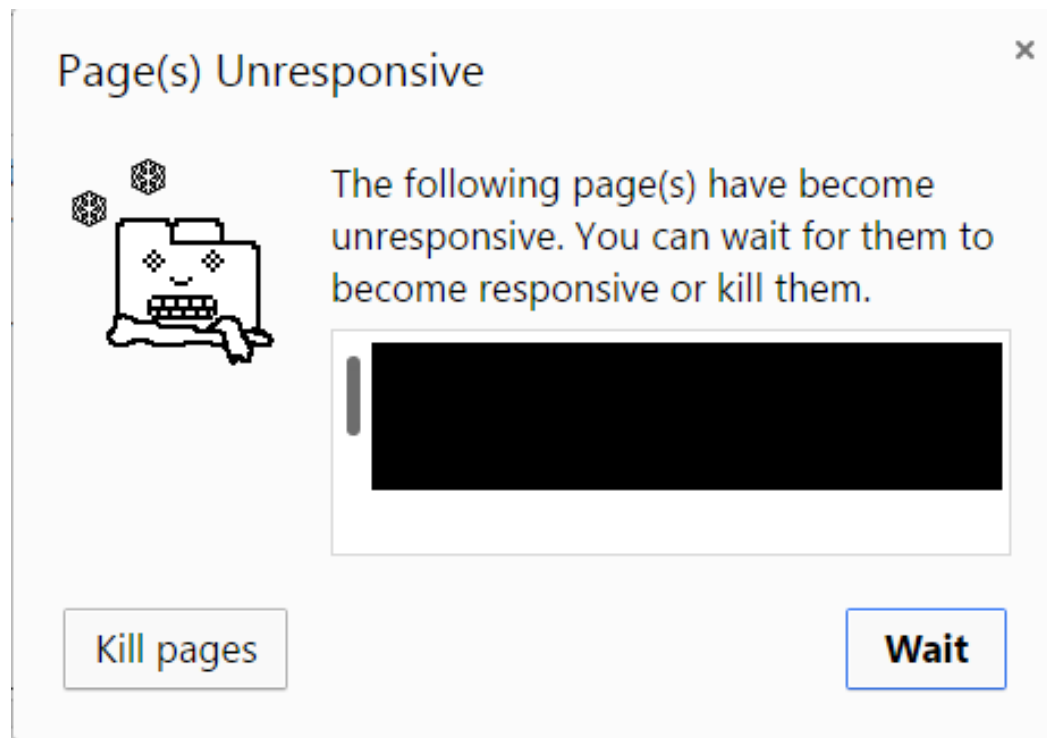


# Single Threaded?

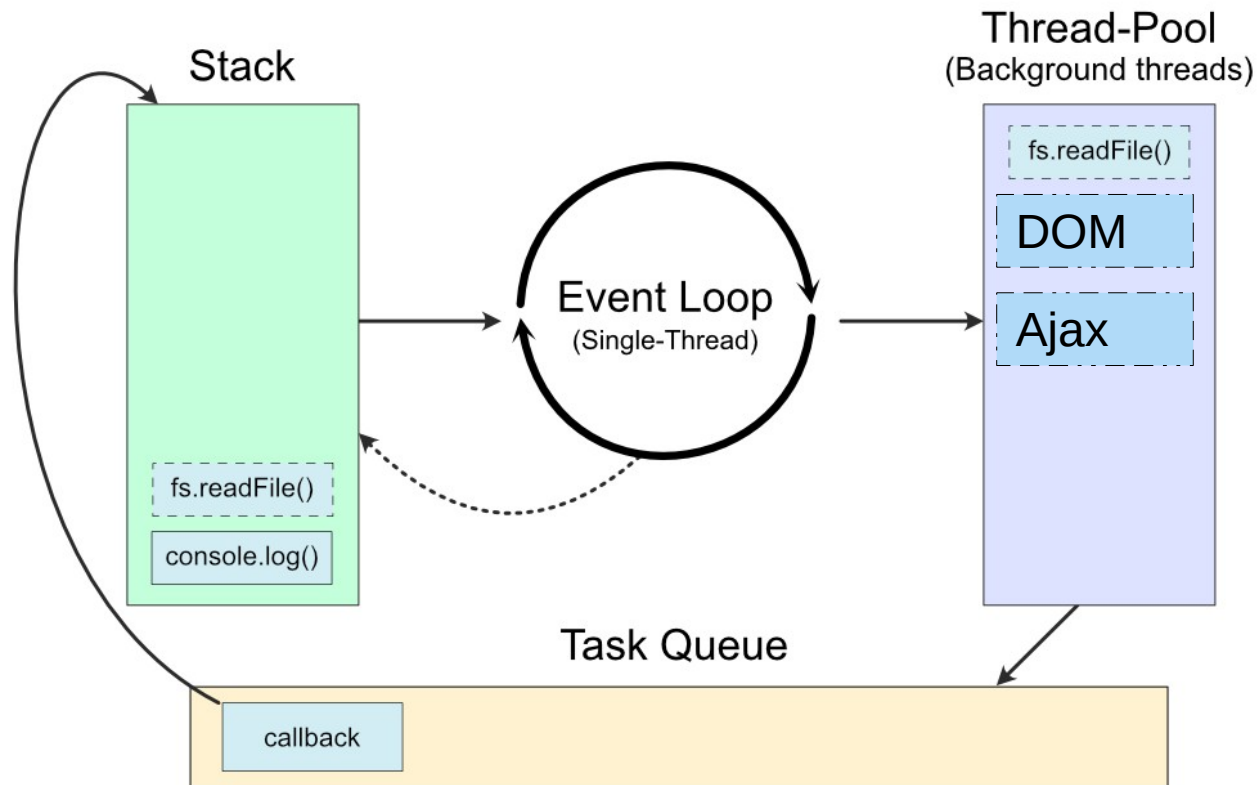
Significa que a execução do código será feita uma parte de cada vez.

Como a execução do código é feita sequencialmente, um código que leve muito tempo executando bloqueará qualquer código que venha depois.

Portanto, às vezes você vê abaixo da tela ao usar o Google Chrome



# Single Threaded? (Event loop)



✓ Event-loop?

✓ Callback?

✓ DOM?

✓ Ajax?

Acima estão alguns conceitos fundamentais para a compreensão do Javascript (e que não vão encontrar paralelo em outras linguagens)



# Tipagem Dinâmica

## Tipagem dinâmica vs tipagem estática:

Java: `int i = 10;` (estaticamente tipada)

`i = "Maria";` 🤯

Python: `i = 10` (dinamicamente tipada)

`i = "Maria"` 😊

Javascript: `let i = 10;` (dinamicamente tipada)

`i = "Maria";` 😊

# Função: cidadã de primeira classe

## Uma função em Javascript pode:

- Possuir atributos (dados)

- Declarar outra função internamente

- Ser passada como parâmetro de outra função

- Ser retornada por outra função

- Ser atribuída a uma variável

# Função: cidadã de primeira classe

Atribuir uma função a uma variável:

```
const foo = function() {  
    console.log("foobar");  
}  
// Chamar a função usando a variável  
foo();
```

# Função: cidadã de primeira classe

Passar função como argumento:

```
function sayHello() {  
    return "Hello, ";  
}  
function greeting(helloMessage, name) {  
    console.log(helloMessage() + name);  
}  
// Passar `sayHello` como um argumento pra função `greeting`  
greeting(sayHello, "JavaScript!");
```

# Função: cidadã de primeira classe

Retornar uma função:

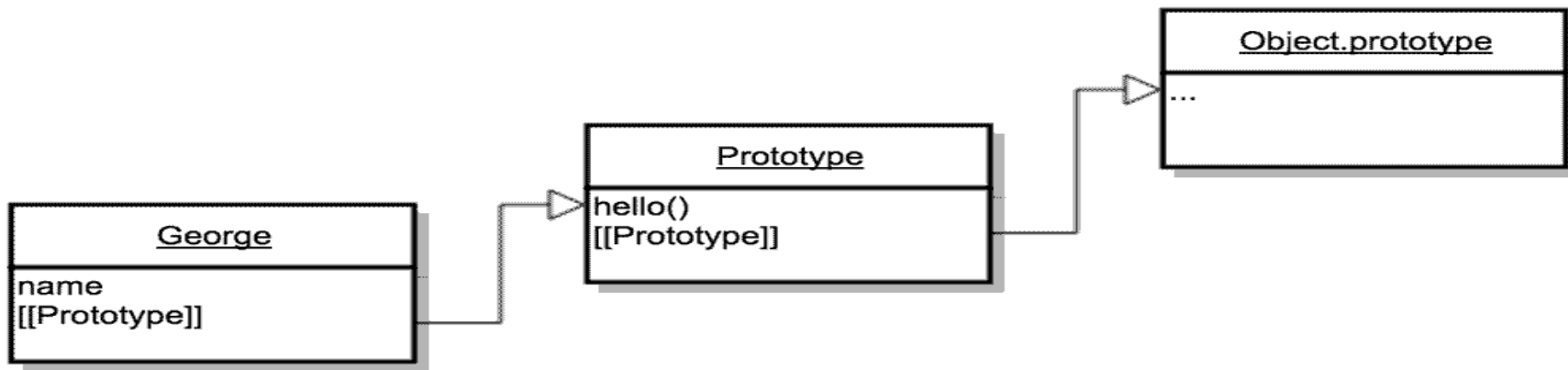
```
function sayHello() {  
  return function() {  
    console.log("Hello!");  
  }  
}
```



```
const sayHello = function() {  
  return function() {  
    console.log("Hello!");  
  }  
}  
const myFunc = sayHello();  
myFunc();
```

```
function sayHello() {  
  return function() {  
    console.log("Hello!");  
  }  
}  
sayHello()();
```

# Herança por protótipo



- Um objeto “herda” de outro objeto – e não de uma classe.
  - Para ser mais exato o mecanismo correto seria *delegação* e não herança.
- Para resumir, é uma implementação diferente das principais linguagens orientadas a objeto, que adotam a herança baseada em classes – como Java, Python, Go, Swift, etc.

# Hoisting (içamento)

//a declaração da variável com **var** é içada (hoisted)  
console.log(forma); // Saída (Output): undefined  
var forma = "quadrado";  
console.log(forma); // Saída (Output): "quadrado"

//a declaração da variável com **let** não é içada  
console.log(forma); // Saída (Output): Erro!!!!  
var forma = "círculo";  
console.log(forma);

```
console.log(forma);  
var forma = "quadrado";  
console.log(forma);
```

undefined

quadrado

```
console.log(forma2);  
let forma2 = "quadrado";  
console.log(forma2);
```

✖ ▶ Uncaught ReferenceError: forma2 is not defined  
at <anonymous>:2:13